Final Project Report

Author

Abishek Vinodh
21F1005989
(21f1005989@student.onlinedegree.iitm.ac.in)
Myself from Bangalore, currently pursuing a secondary degree in B.Tech. Very passionate about technology and photography, and always keen and hungry to learn new things

Description

The project for MAD-1 was focussed on creating a fully functional website using Python, running on Localhost on the local machine of the user.
The functionality of the website is to reflect the application of a Physical Flashcard virtually which would be programmed to behave so

For the login page, I kept it simple without adding a layer of authentication/verification or password hashing in the Database. It uses a HTML form and posts it in the server which is then corresponded against a registered user with the same username or password and logs the individual in, else it posts an error accordingly(Wrong password, not registered...)

Coming to the dashboard functionality, we have to render a webpage dashboard containing the individual's decks. The database in the backend supports all the storage of users and all the cards they wish to create and also storing all the flashcards within each of the decks.
This was practically simple, since it was mostly a HTTP GET request.

Dealing with the Deck Management with Database Management System, I'd Used SQLite for DBMS and SQLite3 to connect the Database with the server. Since this was familiar with DBMS, it was rather easy to integrate the connection. The python commands the Database to create tables and its corresponding sequences and attributes.

Finally, coming to the review system , the user is supposed to select one of the decks for which he/she wants to review. The server then shuffles the corresponding cards to the user one by one, the user will have to guess the answer of each flashcard and post a self review (easy, medium or hard) based on individual's best interests
These reviews shall reflect in the database against the user's scoring.

Technologies Used:

- from flask import Flask - Importing the main Flask module, Flask constructor takes the name of the current module (__name__) as argument.
- from flask import render_template - Flask function used to generate output from a template file based on the Jinja2 engine that is found in the application's templates folder
- from flask import request - The Flask request object contains the data that the client (eg a browser) has sent to your app - ie the URL parameters, any POST data, etc

- from flask import redirect - used to redirect to different views
- from flask import url_for - url_for generates a URL to an endpoint using the method passed in as an argument
- from flask import session - The Session is the time between the client logs in to the server and logs out of the server
- import sqlite3 - used to perform functions like create, read , update and delete database
- from datetime import datetime - used to get current time

Database Schema Used:

1. Table **Cards**: card_id (primary key, Auto-increment)
   Deckname (Not nullable)
   Created on
   Front.
   Back
   Score (Default = 0)
   Last_rev

2. Table **Users:** id (Primary key, Auto-increment)
   Name
   Username (Not nullable, unique)
   Password
   date_created

Architecture and Features:

The main python files used are stored as "**main.py" -** where the actual integration and calling of the server takes place and **"create-db-table" -** where the files proceeds to create a DB with the above schema
The rest of the HTML files are stored in the template file, separately for each individual webpage

Feature of the Website:
The website when hosted locally redirects to login page where the user can enter login details if the username is not matched in the database it redirects to registration page where the user can register and the it goes to the Dashboard where the user can Add, Delete, Update, Review deck and they can also add card to the the deck after which they will review it.

Link to video/screencast:

https://drive.google.com/drive/folders/1PoiGCBxhhibME_YKDnJmOzwqKEdATjX-?usp=sharing